CS 7775

Seminar in Computer Security: Machine Learning Security and Privacy

Fall 2023

Alina Oprea
Associate Professor
Khoury College of Computer Science

November 20 2023

Adversarial Machine Learning: Taxonomy

Attacker's Objective

Learning Stage

	Integrity Target small set of points	**Availability** Target entire model	**Privacy** Learn sensitive information
Training	Targeted Poisoning Backdoor Poisoning Subpopulation Poisoning	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks	Sponge Adversarial Examples	Reconstruction Membership Inference Model Extraction Property Inference

Pang et al. On the Security Risks of AutoML.

USENIX Security 2023

Background

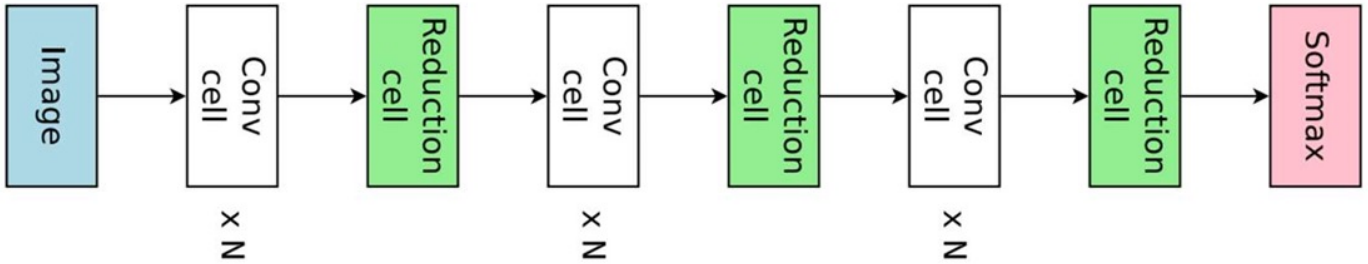
- Automated Machine Learning (AutoML)
  - Auto Data Augmentation
  - Hyperparameter Optimization
  - **Neural Architecture Search (NAS)**
  - etc.

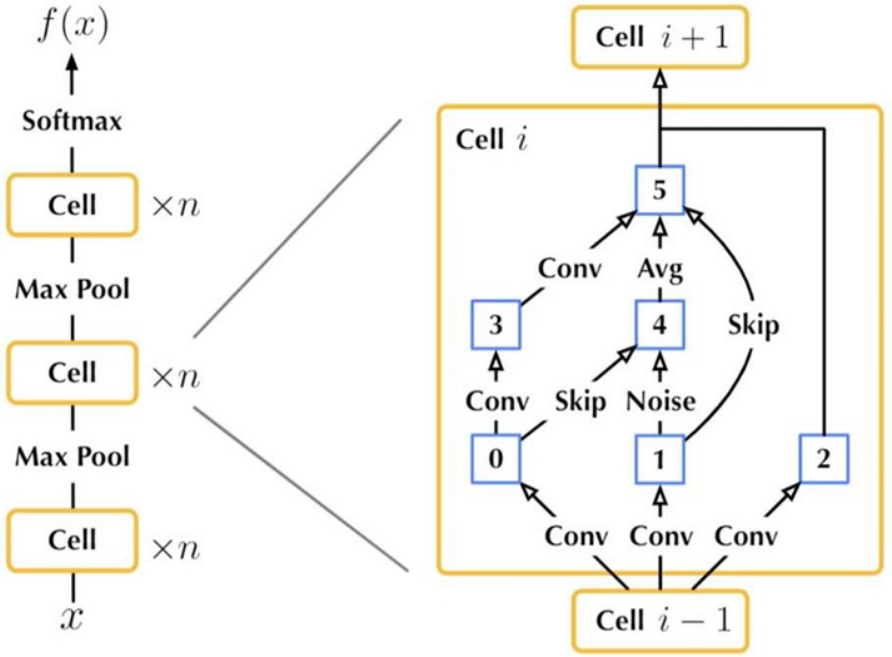
Google's AutoML

AutoGluon

Background

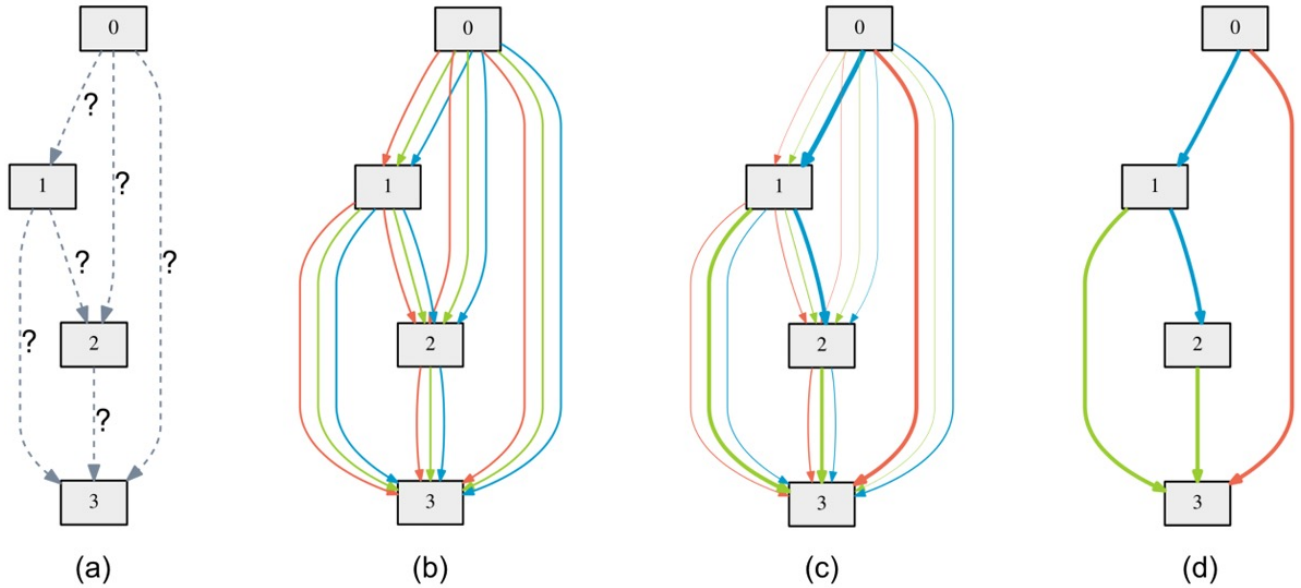
- Neural Architecture Search (NAS)
  - NAS searches good architectures automatically.
  - Differential Architecture Search (DARTS)
    - Efficient
    - Cell-based





- Architecture is a stack of cells
- Each cell is a DAG with multiple operations (learned)

DARTS



$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)})$$

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

Bilevel optimization

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$
$$\text{s.t.} \quad w^*(\alpha) = \text{argmin}_w \ \mathcal{L}_{train}(w, \alpha)$$

Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

Optimize jointly for architecture and parameters

- Introduce continuous weights on edges α
- Operation on edge has maximum weight

DARTS

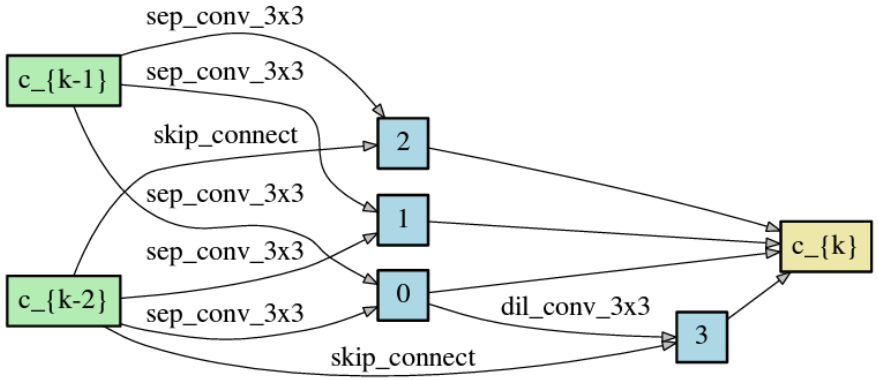


Figure 4: Normal cell learned on CIFAR-10.

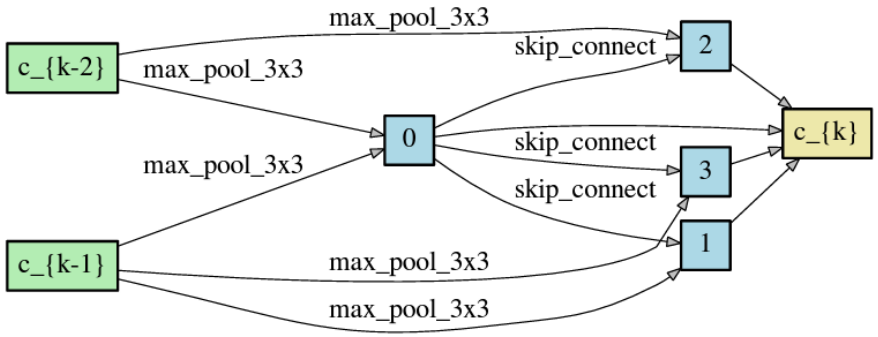


Figure 5: Reduction cell learned on CIFAR-10.

Problem Statement

- Is AutoML susceptible to various adversarial attacks at training and testing time?
- How do AutoML architectures compare to manual architectures in terms of robustness?
- How to increase robustness?

Threat Model

- Evasion attacks
  - White-box: PGD

$$\min_{\delta \in \mathcal{B}_\varepsilon} \ell(f(x+\delta), t)$$

- Poisoning availability
  - Maximize model loss

$$\max \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{tst}}} \ell(f_{\theta^*}(x), y)$$
$$\text{s.t. } \theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{trn}}\cup\mathcal{D}_{\text{pos}}} \ell(f_\theta(x), y)$$

- Backdoor poisoning

$$\min_{r\in\mathcal{R}_\varepsilon,\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{trn}}} [\ell(f_\theta(x), y) + \lambda\ell(f_\theta(x+r), t)]$$

- Functionality stealing
- Membership inference MI: confidence or label-only

Datasets / Models

	Architecture	CIFAR10	CIFAR100	ImageNet32
Manual Architecture	*BiT* [32]	96.6%	80.6%	72.1%
	DenseNet [28]	96.7%	80.7%	73.6%
	DLA [60]	96.5%	78.0%	70.8%
	ResNet [26]	96.6%	79.9%	67.1%
	ResNext [57]	96.7%	80.4%	67.4%
	VGG [52]	95.1%	73.9%	62.3%
	WideResNet [61]	96.8%	81.0%	73.9%
NAS Architecture	*AmoebaNet* [47]	96.9%	78.4%	74.8%
	DARTS [39]	97.0%	81.7%	76.6%
	DrNAS [11]	96.9%	80.4%	75.6%
	ENAS [46]	96.8%	79.1%	74.0%
	NASNet [64]	97.0%	78.8%	73.0%
	PC-DARTS [59]	96.9%	77.4%	74.7%
	PDARTS [12]	97.1%	81.0%	75.8%
	SGAS [35]	97.2%	81.2%	76.8%
	SNAS [58]	96.9%	79.9%	75.5%
	Random [17]	96.7%	78.6%	72.2%

Evasion Results

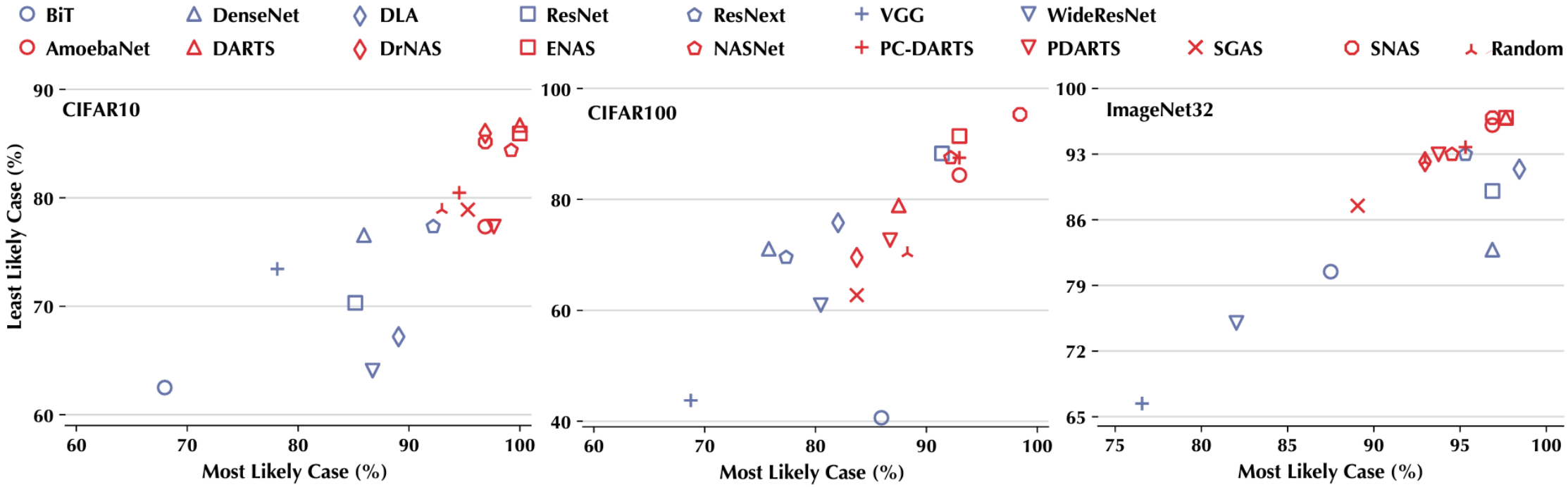


Figure 2: Performance of adversarial evasion (PGD) against NAS and manual models under the least and most likely settings.

Evasion Results

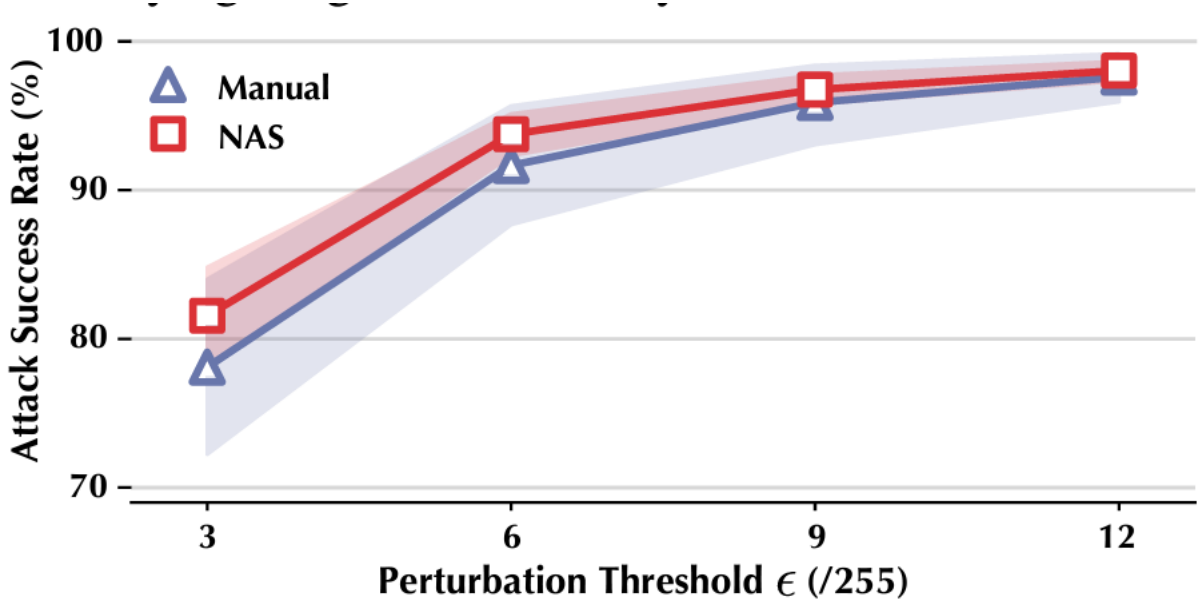


Figure 3: Impact of perturbation threshold (ε) on the vulnerability of different models with respect to PGD on CIFAR10.

Poisoning Results

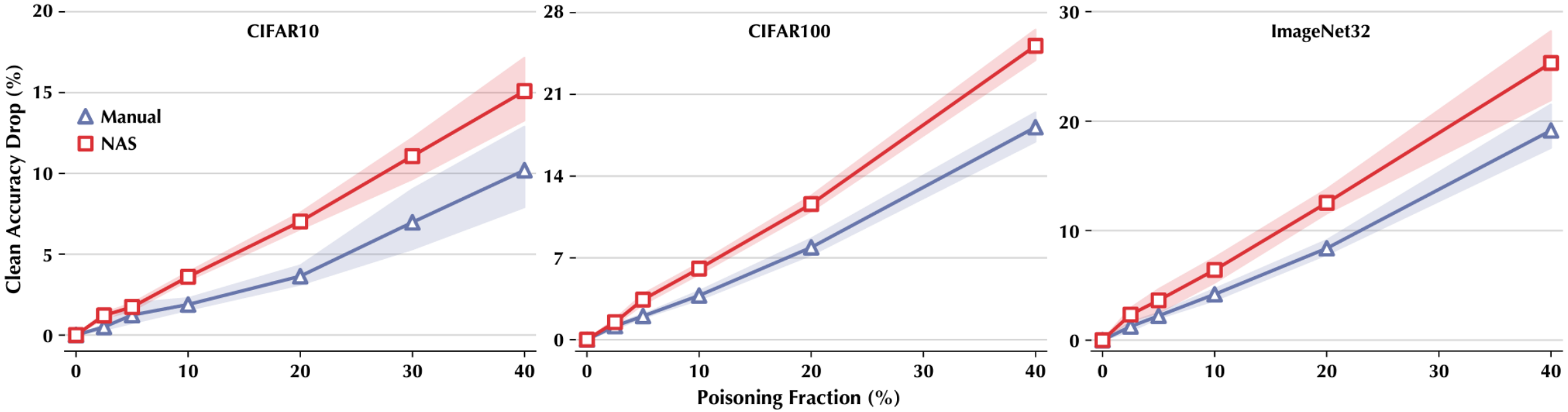
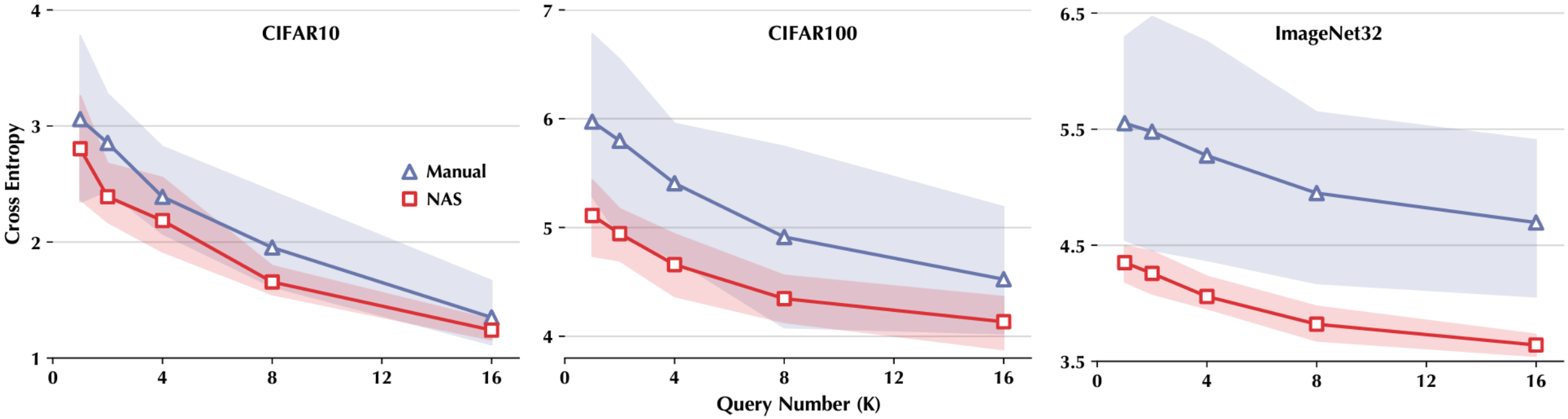


Figure 6: Performance of model poisoning against NAS and manually designed models under varying poisoning fraction p_{pos}.

Functionality Stealing



Label-Only Membership Inference

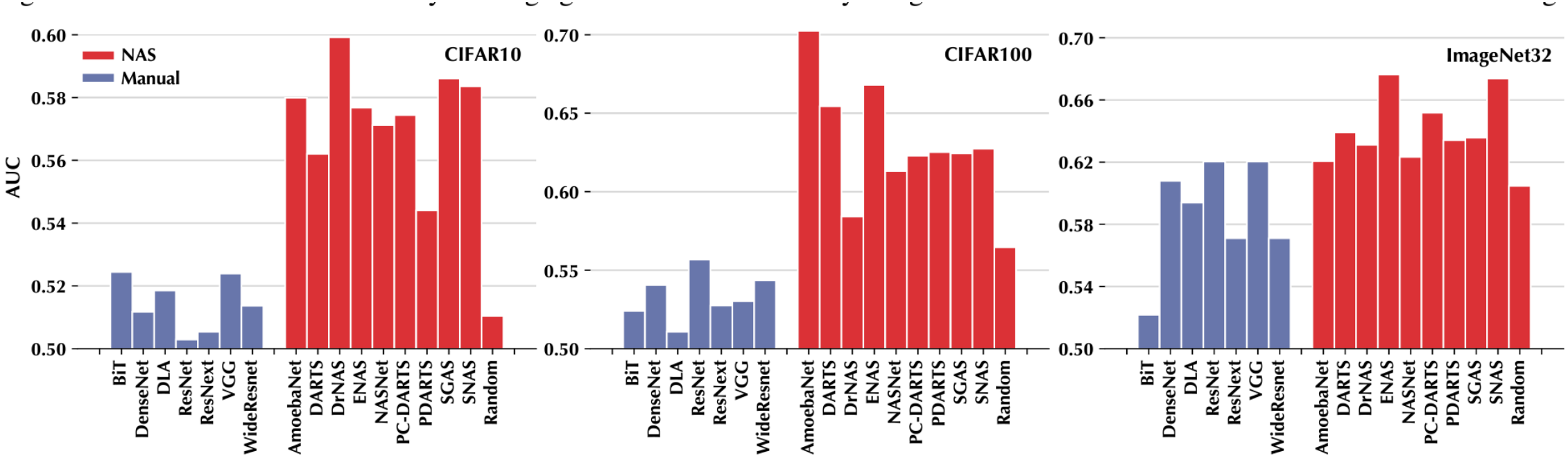


Figure 10: Performance of label-only membership inference attacks against NAS and manually designed models.

Why Higher Vulnerability?

Hypotheses

High loss smoothness – *The loss landscape of NAS models tends to be smooth, while the gradient provides effective guidance for optimization. Therefore, NAS models are amenable to training using simple, first-order optimizers.*

Low gradient variance – *The gradient of NAS models with respect to the given distribution tends to have low variance. Therefore, the stochastic gradient serves as a reliable estimate of the true gradient, making NAS models converge fast.*

Loss Smoothness

Loss smoothness – A loss function $\mathcal{L}$ is said to have L-Lipschitz ($L > 0$) continuous gradient with respect to θ if it satisfies $\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\| \leq L\|\theta - \theta'\|$ for any θ, θ'. The constant L controls $\mathcal{L}$'s smoothness. While it is difficult to directly measure L of given model f, we explore its loss contour [22], which quantifies the impact of parameter perturbation on $\mathcal{L}$. Specifically, we measure the loss contour of model f as follows:

$$\Gamma(\alpha, \beta) = \mathcal{L}(\theta^* + \alpha d_1 + \beta d_2) \tag{12}$$

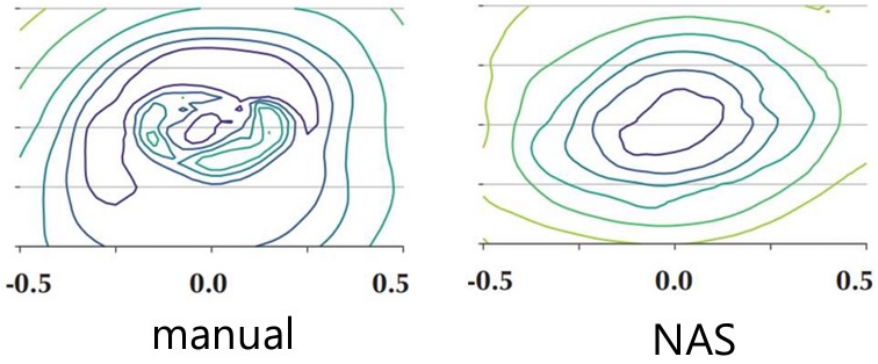
where θ^* denotes the local optimum, d_1 and d_2 are two random, orthogonal directions as the axes, and α and β represent the perturbation steps along d_1 and d_1, respectively. Notably, the loss contour effectively approximates the loss landscape in a two-dimensional space [36].

Analysis

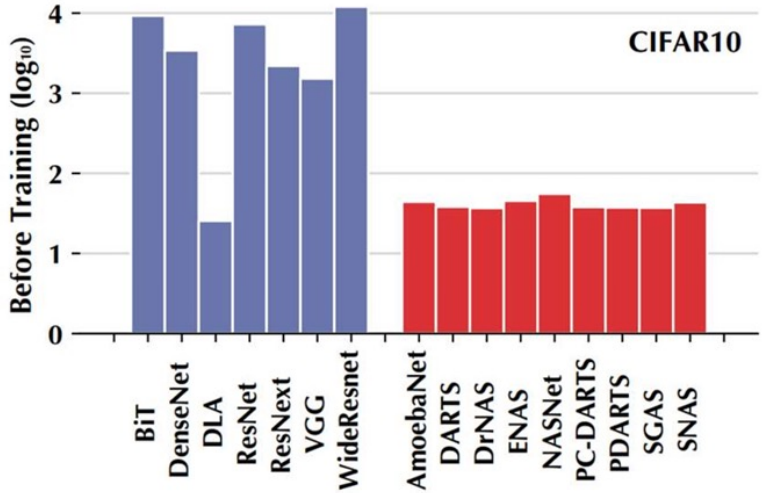
- NAS algorithms prefer architectures that converge fast.
  - Shallow models
  - More skip connects

⟹ NAS model characteristics:

- High Loss Smoothness (small Lipschitz constant)

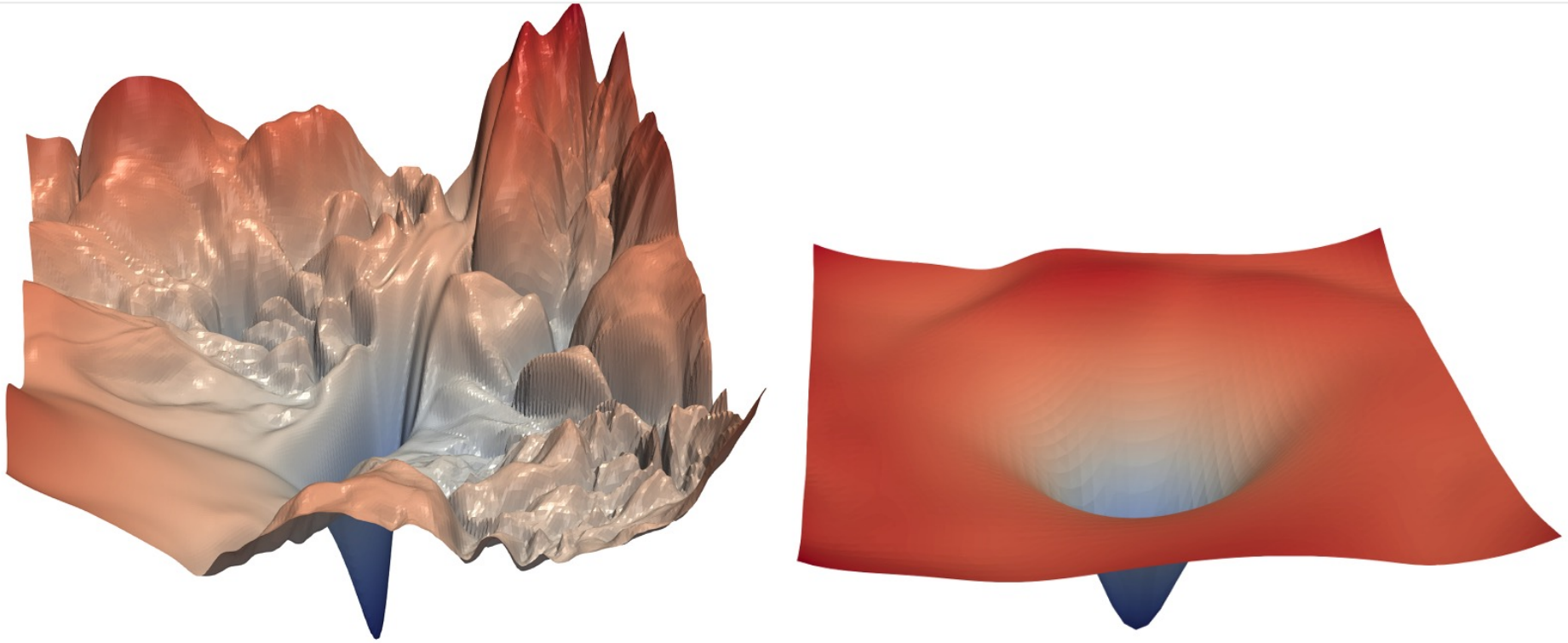


- Low gradient variance



Conclusion: More accurate gradients lead to higher vulnerability to attacks based on gradients (evasion, poisoning, stealing, MI)

Li et al. Visualizing the Loss Landscape of Neural Nets



(a) without skip connections

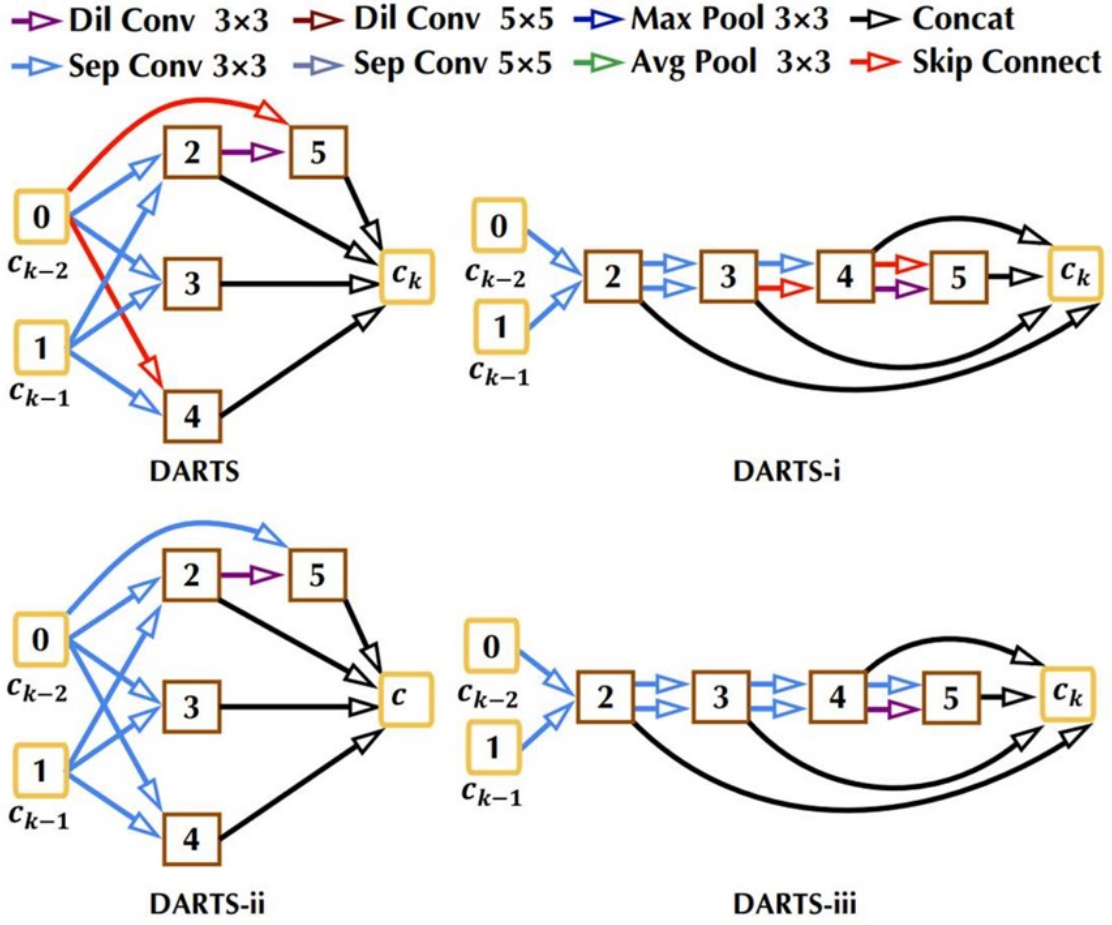
(b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

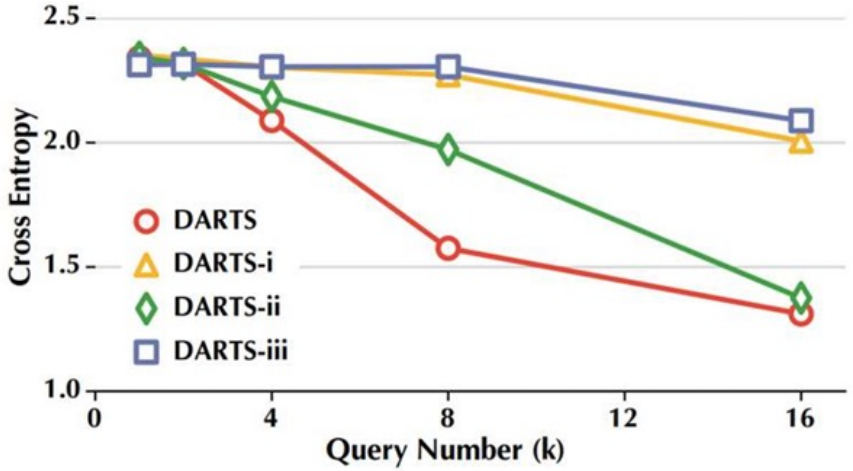
Mitigations

- To suppress those characteristics,
  - (i) increase cell depth
  - (ii) reduce skip connects
  - (iii) combined of (i) and (ii)

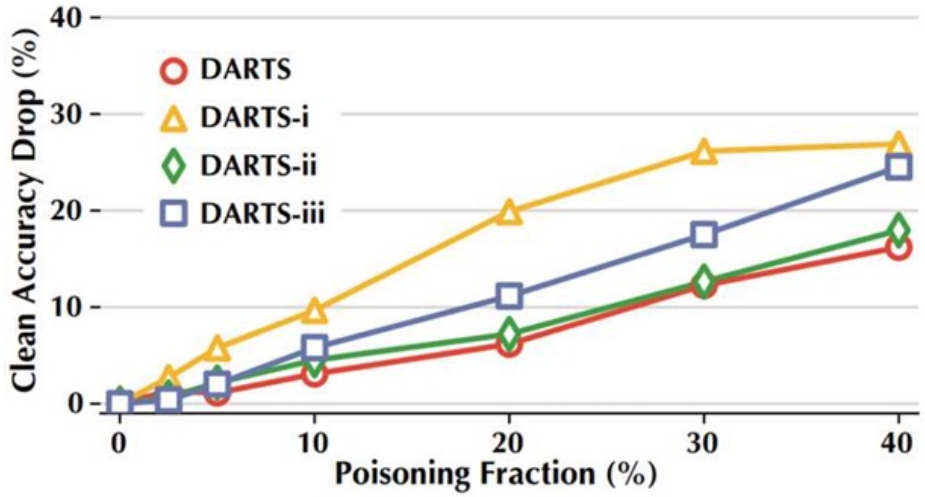


Mitigation Evaluation

- Functional Stealing



- Model Poisoning



Summary

- Strengths
  - Evaluated 5 different types of attacks
  - Considered 9 AutoML methods and compared against well-known baseline architectures on 3 datasets
  - Analysis of loss smoothness and gradient variance is interesting
- Limitations
  - Mitigations are not effective for all attacks
  - Security of AutoML deserves further investigation
    - Add robustness objectives during training
- Acknowledgement to the paper authors for their slides